APIの傾向によるラベル付けとSVMによるマルウェアの分類

碓井 利宣† 重松 邦彦‡ 武田 圭史† 村井 純†

†慶應義塾大学環境情報学部
255-0882 神奈川県藤沢市遠藤 5322
{alc, keiji, jun}@sfc.wide.ad.jp

‡慶應義塾大学大学院政策・メディア研究科
255-0882 神奈川県藤沢市遠藤 5322
sigematu@sfc.wide.ad.jp

あらまし インターネット利用の普及に伴い，様々な悪意を持った新たなマルウェアが日々出現しており，これらについて効果的な対応を効率よく実施するためには，発見されたマルウェアを短時間で分析する必要がある．本研究では，マルウェアの静的解析手法を用いてマルウェアの挙動に関する情報を抽出し，そこで利用されるAPIの傾向によってラベル付けを行う．それらの情報を基にして機械学習であるSupport Vector Machineにより分類する．本手法によって特に挙動の類似性の高いマルウェア同士が同じグループとして分類するシステムを実装した．本システムを用いることで，分析者は分類結果から挙動の傾向を短時間で把握することができ亜種の特定や対策の立案などに活用できる．

Malware Classification based on SVM and Labeling by API's Tendency

Toshinori Usui† Kunihiko Shigematsu‡ Keiji Takeda† Jun Murai†

†Faculty of Environment and Information Studies, Keio University
5322 Endo, Fujisawa 255-0882, JAPAN
{alc, keiji, jun}@sfc.wide.ad.jp

‡Graduate School of Media and Governance, Keio University
5322 Endo, Fujisawa 255-0882, JAPAN
sigematu@sfc.wide.ad.jp

Abstract With the spread of the internet, various kinds of new malwares have appeared. Therefore, to take effective measures to cope with these efficiently, we need to analyze malwares in fast method. In our proposing method, we extract information which is related to malwares' behavior by static analysis, affix labels to malwares based on its APIs' tendency, and classificate them by Support Vector Machine. We implemented automatic classification system according to our method. By using our system, analysts can know malwares' behavior easily from the result of the classification.

1 研究の背景

近年，IPA（情報処理推進機構）や，ウイルス対策ソフトを開発している企業の調査において，数多くのマルウェアが報告されている．それによって，手作業による解析は限界を迎えていると言える．また，特にユニークなマルウェアが増加しており，ヒューリスティック手法が重要化している．さらに，パッカーや自動でマルウェアを生成するツールなどにより亜種の作成

が容易となっており，マルウェアが大量に出現する要因も多く，今後も増加していくことが考えられる．本稿では，マルウェアを挙動に基づいて自動で分類する手法を提案し，さらに，挙動を基にして付与したラベルを利用して分類することにより，分類結果から挙動の方向性を把握出来るようにする手法も提案する．また，提案手法を基に構築したマルウェア分類システムについて述べる．これらによって，自分の手で解析せずに挙動の大枠が分かることによる研究活動の効率化や，マルウェアの知識が無くても研究に適した検体を選択出来ることによる隣接分野の研究者の支援などが望まれる．

2 先行研究

マルウェアの API に着目して分類を行う研究は多く存在する．それらでは API の種類のみでなく，API のコールツリーを構成して用いるなど，様々な工夫が施されている．一方，分類対象のマルウェアについて，ラベルのように分類結果に対して分かりやすく工夫した形で情報を付加している研究はあまり見られない．また，分類の際に利用する手法としては，Neural Network やベイズ学習を始めとして非常に多様な手法が用いられてきている．その中でも，SVM は高い精度を持ち，教師信号による学習時の入力次元が学習データ量よりも非常に大きい場合でも正しく認識しやすいという特徴もあるため，API を基にしたマルウェアの分類には適していると言える．一方，入力次元が大きくなるに連れて計算量が増加するという特徴も持つため，入力データの次元が大きくなりすぎないよう工夫する必要がある．

3 提案手法

提案するマルウェアの分類手法について述べる．本手法ではまず，分類に先立って事前に複数のマルウェアのクラスタリングを行い，これによって生まれるクラスタを分類先となるクラスとして定義する．そして、このクラスタリングの結果を教師信号として学習器に入力し，モデルをビルドしておき，新たに分類対象となるマルウェアが出現した際にはそれを用いてクラスに分類していく．この手法に必要な情報抽出，ラベル付与，クラスタリング，分類の 4 つの行程について，それぞれ以下に詳細を示す．

3.1 情報抽出

マルウェアの PE ヘッダを参照し，内部構造に関する情報を得る．それを基にして，マルウェアの挙動に関する情報を抽出する．抽出する情報は，具体的には以下の 2 つである．1 つ目はインポートセクションを参照することで得られる，マルウェアがインポートしている DLL, API の情報である．また，2 つ目はコードセクションを参照することで得られる，far CALL 命令による API 呼び出しの情報である．これらの情報を後に行うラベルの付与やクラスタリング，分類の際に利用する．

3.2 ラベル付与

抽出した API の情報を基にして，マルウェアに対して動作の傾向に関するラベルを付与する．ラベルを付与する手法については，以下のようになる．まず，あらかじめ API の動作の方向性ごとに，API の種類と付与するラベルの対応を定義しておく．そして，前節の手法によってラベルを付与したいマルウェアから抽出したインポートしている API に関する情報をこの定義と照らし合わせ，マルウェアのインポートしている API に対応したラベルを列挙する．それらを集計し，動作が特に顕著に表れているものをマルウェアを特徴付けるラベルとして付与する．ある検体のインポートしている API とそれに対応するラベルを表 1 に示す．

3.3 クラスタリング

挙動を基にマルウェアのクラスタリングを行う．前節で付与した挙動の傾向によるラベルを基に，近いラベルを持つマルウェア同士を同じクラスタにまとめる．ここで作成されたクラス

DLL	API	ラベル
KERNEL32	GetProcAddress	メモリ操作
	LoadLibrary	メモリ操作
	GrobalAlloc	メモリ操作
	VirtualProtect	メモリ操作
	GlobalFree	メモリ操作
	CreateFile	ファイル操作
	ReadFile	ファイル操作
	CloseHandle	ハンドル操作
	CopyFile	ファイル操作
	FreeLibrary	メモリ操作
	GetCommConfig	通信デバイス制御
	ReportEvent	イベントログ操作
ADVAPI32	CopySid	SID 操作
	ConvertStringSidToSid	SID 操作
	IsValidSid	SID 操作
	ConvertSidToStringSid	SID 操作
ole32	CoUninitialize	COM
	GetClassFile	COM
	CLSIDFromProgID	COM
	CoInitialize	COM

表 1: 検体のインポートしている API とラベルの対応

タは分類の際に分類先のクラスとして用いられるため，各クラスにクラス ID を割り振る．

3.4 分類

分類は機械学習によって行う．学習アルゴリズムには，Support Vector Machine（以下，SVM）を用いる．SVM は教師あり学習であり，教師信号を必要とする．本手法では，教師信号には前節の手法で得られるクラスタリング結果を用いる．また，入力データのデータ構造は以下のようなものを想定している．マルウェア検体から API の種類ごとの呼び出し回数を要素とする特徴ベクトルを生成し，これを入力データとする．図.1 に入力データのデータ構造を示す．予測のためのモデルを作成する際には，教師信号として，クラスタリングの結果得られたクラス ID と i 個の検体から得られる x1, x2, …, xi の i 本の特徴ベクトルを利用する．実際に予測を行う際には，上記の手法によって作成されたモデルに対して，分類対象の検体から得られた 1 本の特徴ベクトルを入力し，予測させる．

	API①	API②	API③	API④		API(N)
x_1 = (	1,	2,	0,	1,	…,	2)
x_2 = (	3,	0,	2,	1,	…,	0)
x_3 = (	2,	2,	0,	0,	…,	0)
			⋮			
x_i = (	a_1,	a_2,	a_3,	a_4,	…,	a_n)

図 1: SVM の入力データのデータ構造

4 マルウェア分類システムの構成

提案した手法によって実際にマルウェアの分類を行うシステムの構成を述べる．大きくシステムは事前準備部と分類部の 2 つの部分に分かれる．事前準備部にて複数のマルウェア検体のラベル付けとクラスタリングを行うことで教師信号を作成して SVM の予測モデルをビルドし，分類対象が出現した際には分類部にて SVM による予測を行う．以下に，各部の構成を示す．

4.1 事前準備部

事前準備部は，情報抽出部，ラベル付与部，クラスタリング部，機械学習モデル作成部の 4 つの部分からなる．以下に，各部の詳細を示す．また，図.2 に事前準備部のシステムの構成を示す．

4.1.1 情報抽出部

情報抽出部では，マルウェアからの情報の抽出を行う．まず，マルウェアからメモリダンプを取得した上でインポートテーブルの再構築を行う．PE ヘッダを分析することによってインポートセクションとコードセクションのメモリダンプファイル上の位置を取得し，インポートセクションを参照することで DLL および API を，コードセクションを逆アセンブルすることで API の呼び出しをそれぞれ取得する．

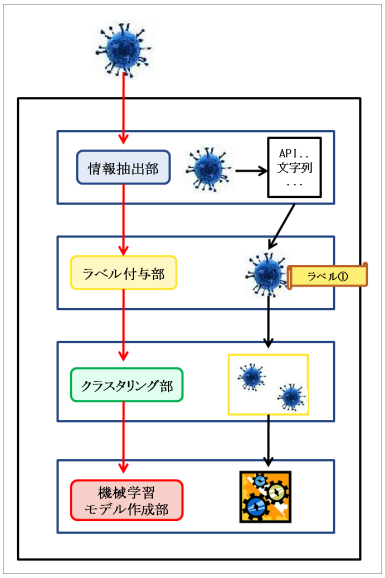

図 2: 事前準備部のシステム構成

4.1.2 ラベル付与部

ラベル付与部では，マルウェアに対してラベルの付与を行う．情報抽出部で取得したマルウェアが利用している API をあらかじめ作成した API の種類とラベルの対応を定義したデータベースに照らし合わせ，特徴的なラベル群をマルウェアに対するラベルとして，データベースに格納する．

4.1.3 クラスタリング部

クラスタリング部では，マルウェアの挙動に関する情報を基にクラスタリングを行う．前述の通り，マルウェアの持つラベルの距離によってクラスタリングを行い，作成されたクラスタにはクラス ID を割り振る．

4.1.4 機械学習モデル作成部

機械学習モデル作成部では，情報抽出部にて取得した API の情報を基にして 3.4 節で提案したデータ構造による入力データを作成し，クラスタリング部で得られた結果のクラス ID とともに教師信号として，SVM によって予測モデルのビルドを行う．

4.2 分類部

分類部は，情報抽出部，機械学習予測部，結果抽出部の 3 つの部分からなる．以下に，各部の詳細を示す．また，図.3 に分類部のシステムの構成を示す．

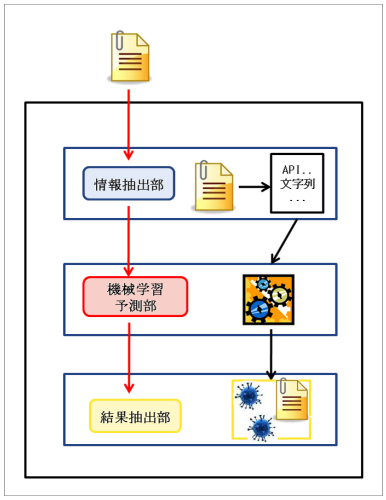

図 3: 分類部のシステム構成

4.2.1 情報抽出部

分類部内の情報抽出部における情報抽出では，分類対象に対して 4.1.1 項で述べた事前準備部内の情報抽出部と同様のことを行う．

4.2.2 機械学習予測部

機械学習予測部は，作成された予測モデルを基に SVM によって分類先のクラスの予測を行う．まず，機械学習モデル作成部と同様にして情報抽出部にて取得した API を基に入力データを作成する．そしてそのデータを事前準備部で作成したモデルに入力し，分類先と予測されたクラス ID を出力として得る．

4.2.3 結果抽出部

結果抽出部では，分類先や分類先から推定されるラベルなどを分類結果としてデータベースに格納する．

5 実験

構築したマルウェア分類システムによって実際に分類を行った．その分類結果とそれに対する考察について述べる．実験対象は慶應義塾大学にてマルウェア収集サーバによって収集した検体を利用した．互いにハッシュ値の異なる検体群から教師信号として 50 検体を，分類対象として 15 検体を無作為抽出し，それらに対して実際に予測モデルの作成と分類を行った．

実権の結果，50 検体から作成されたクラスタは 12 個であり，それらを分類先となるクラスとした場合，15 検体の分類先には 8 個のクラスが該当した．各クラスにはそれぞれバックドア型のトロイの木馬，ダウンローダ型のトロイの木馬，オンラインゲームのパスワード窃盗型トロイの木馬，ファイルの破壊を行うドロッパー型のトロイの木馬，スケアウェア，アドウェア，スパイウェアが含まれていた．分類結果をクラスに含まれるマルウェアの特徴とそのクラスに分類されたマルウェア数にまとめて表 2 に示す．亜種であるのに別のクラスとして分類された検体も存在した一方，クラス内において，挙動のまったく異なったマルウェアが同じクラスに分類された例は今回の実験では見られなかった．結果より，15 検体中 10 検体が正しく分類されたと考えられた．分類の際に, 複数の性質を併せ持つ検体が含まれていた場合，その性質群のうちの単一の性質を持つ検体とは別のクラスとして処理されると考えられる．例えば，ダウンローダ型のトロイの木馬とバックドア型のトロイの木馬の両方の性質を併せ持つ検体が分類対象として存在する場合，どちらの性質を持つ検体が含まれるクラスにも分類されず，全く別のクラスが作成され，そのクラスに分類されると考えられる．より挙動や性質を知ることを重視した分類を行うためには，複数の性質を持つ検体のクラスとそれらのうちの単一の性質を持つ検体のクラスとを結びつける必要があり，階層化などを検討している．

ラベル付けについては，スケアウェアで構成されたクラスに分類された検体である FakeAlert に，ファイル操作，プロセス操作，ウインドウ操作，環境変数操作，レジストリ操作，ネットワークなどのラベルが付与されていることに対して，実際にその FakeAlert の挙動に実行中のプロセスのリストアップ，開いているウインドウのリストアップ，Windows フォルダへの実効ファイルの書き込み，環境変数パスの改変，レジストリ要素の追加および変更，http によるネットワーク接続などがあることから，ラベルが挙動の傾向を知る働きを果たしていることが分かる．

クラスの特徴	検体数
バックドア型トロイ	3
ダウンローダ型トロイ	3
オンラインゲームのパスワード窃盗型トロイ	2
ドロッパー型トロイ 1	2
ドロッパー型トロイ 2	1
スケアウェア	1
アドウェア	1
スパイウェア	2

表 2: 分類結果

6 まとめ

本稿では，機械学習によってマルウェアを分類するとともに API の傾向によってラベル付けを行うことで，分類結果からマルウェアの挙動の傾向を知ることができる分類手法を提案し，それを用いたシステムの実装と実験について述べた．実験の際の分類結果は 15 検体中 10 検体が実際のマルウェアの挙動と一致する分類先に含まれていた．また，ラベルを付与することで，分類結果から大まかな挙動の方向性を簡単に知ることができた．

今後は，ラベルの付与およびクラスタリングの手法の洗練を目指すとともに，API 以外の情報をあわせて用いることにより，分類精度の向上や API に関する情報を抽出できない場合でも挙動の傾向を知ることができるような手法を模索していく．

参考文献

[1] G data malware report. http://www.gdata.co.jp/files/2010lasthalfMalwareReport.pdf, February 2011.

[2] Askan Sami, Badak Yadegari, Hossein Rahimi, Naser Peiravian, Sattar Hashemi, Ali Hamzeh. Malware detection based on mining API calls. *SAC 2010*, March 2010.

[3] 岩本誠, 伊藤光恭, 村岡洋一. 機械語命令列の類似性に基づく自動マルウェア分類システム. **情報処理学会論文誌** *51(9)*, September 2010.

[4] Konrad Rieck, Thorsten Holz, Carsten Willems, Patrick Duessel, Pavel Laskov. Learning and Classification of Malware Behavior. *Lecture Notes in Computer Science*, Volume 5137/2008 2008.

[5] Tesauro,G.J.; Kephart,J.O.; Sorkin,G.B. Neural networks for computer virus recognition. *IEEE Expert 11 Issue:4*, August 1996.

[6] 戸部和洋, 森達哉, 千葉大紀, 下田晃弘, 後藤滋樹. 実行ファイルに含まれる文字列の学習に基づくマルウェア検出方法. *MWS2010*, October 2010.

[7] 小池竜一, 中谷直司, 萩原由香里, 厚井裕司, 高倉弘喜, 吉田等明. ベイズ学習アルゴリズムを用いた未知のコンピュータウイルス検出手法. **情報処理学会論文誌** *46(8)*, August 2005.

[8] Huihuan Wang, Naoshi Nakaya, Ryuiti Koike, Dengfeng Zhang, Abdullah Mamun. A Performance Evaluation of Bayes Learning Algorithm For Spam Filter and Virus Filter. *International Conference on Computer, Control & Communication*, 2007.

[9] Paul Graham. Better Bayesian Filtering. *Spam Conference 2003*, 2003.

[10] Gary Robinson. Statistical Approach to the Spam Problem. *Linux JOURNAL*, 2003.

[11] Thorsten Joachins. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, 1997.

[12] Ollydbg. http://www.ollydbg.de.

[13] Importstudio. http://www.reversinglabs.com/forum/index.php?board=18.0.

[14] 碓井利宣, 重松邦彦, 水谷正慶, 武田圭史, 村井純. 機械語命令列の類似度分析を用いた不正コードの分類. **情報処理学会第 *73* 回全国大会予稿集**, March 2011.